TREK

HOW TO BUILD A VISUAL BASIC DISPLAY

TUTORIAL

November 2012

Approved for Public Release; Distribution is Unlimited.

TABLE OF CONTENTS

FIGURES

1   What You Need To Know Before Reading This Document

This tutorial assumes the following:

➢ You are familiar with the material in the TReK Getting Started User Guide (TREK-USER-001) and the TReK Telemetry Tutorial (TREK-USER-002).

➢ You are familiar with the following material in the TReK Telemetry Application Programming Interface Reference Manual (TREK-USER-027):
✶ Sections 1 – 8
✶ GetOneNewestConvertedIntegerValue Function Description

➢ You have some experience with Microsoft Visual Basic 6.0.

➢ You know how to start the TReK Telemetry Processing application, add a packet to the packet list, and activate the packet. (See TReK Telemetry Processing User Guide TREK-USER-003.)

➢ You know how to start the TReK Training Simulator application, add a packet to the packet list, and send the packet. (See TReK Training Simulator User Guide TREK-USER-004.)

If you are uncomfortable with any of the items listed above, some of the terminology and concepts presented in this tutorial may be difficult to understand.

2   Technical Support

If you are having trouble installing the TReK software or using any of the TReK software applications, please try the following suggestions:

Read the appropriate material in the manual and/or on-line help.

Ensure that you are correctly following all instructions.

Checkout the TReK Web site at http://trek.msfc.nasa.gov/ for Frequently Asked Questions.

If you are still unable to resolve your difficulty, please contact us for technical assistance:

TReK Help Desk E-Mail, Phone & Fax:

E-Mail: trek.help@nasa.gov
Telephone: 256-544-3521 (8:00 a.m. - 4:30 p.m. Central Time)
Fax: 256-544-9353

TReK Help Desk hours are 8:00 a.m. – 4:30 p.m. Central Time Monday through Friday. If you call the TReK Help Desk and you get a recording please leave a message and someone will return your call. E-mail is the preferred contact method for help. The e-mail message is automatically forwarded to the TReK developers and helps cut the response time.

3   Introduction

This tutorial will walk you through the process of building a Visual Basic application that displays telemetry data. The TReK Application Programming Interface (API) will be used to retrieve one new telemetry value once every second. The application is called VBDisplay and is shown in Figure 1.
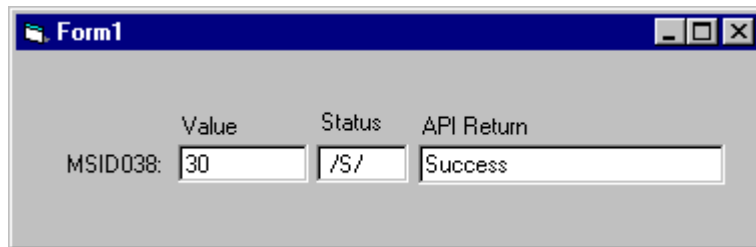
Figure 1 VBDisplay

4   Step-By-Step Instructions

1. Start the Visual Basic application.

2. You will be prompted to choose from three tabs (New, Existing, or Recent) on the opening screen. Choose **New** and **Standard EXE** and push **Open** as shown in Figure 2. If you have clicked the 'Don't show this dialog in the future' box in a previous session, this dialog will not be presented. If this is the case, you can go to the **File** menu and select **New** to open a new project file.
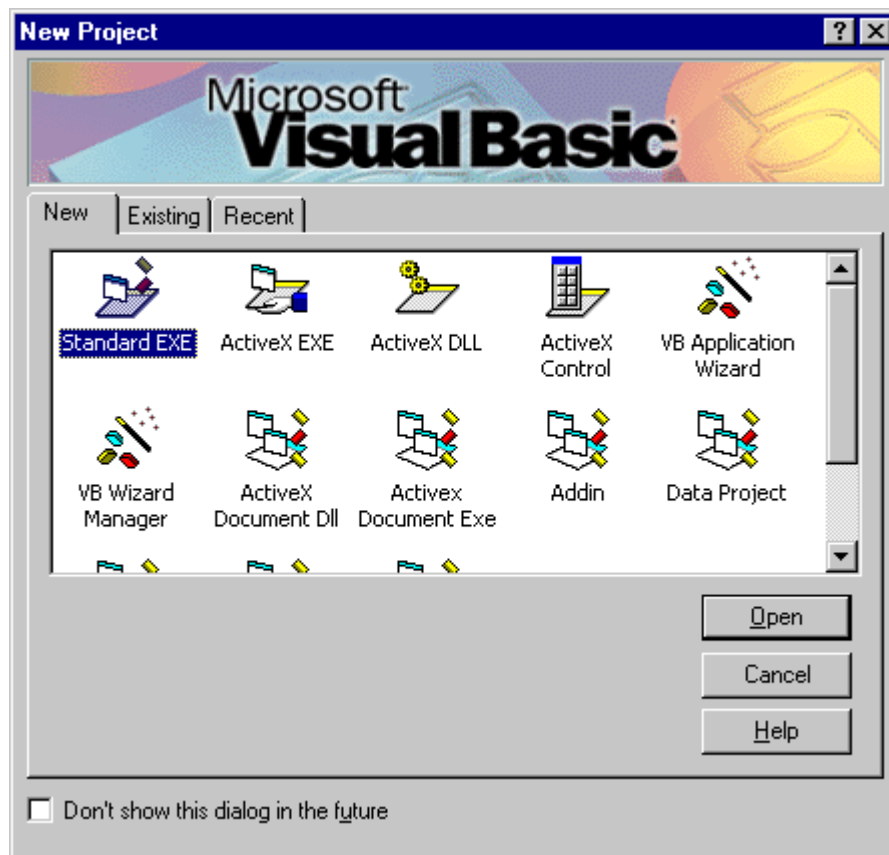
Figure 2 New Project Dialog

3. When Visual Basic creates the new project you will see a window similar to the one shown in Figure 3. During the next few steps you will be adding controls to the form using the tool palette on the left side of the window.
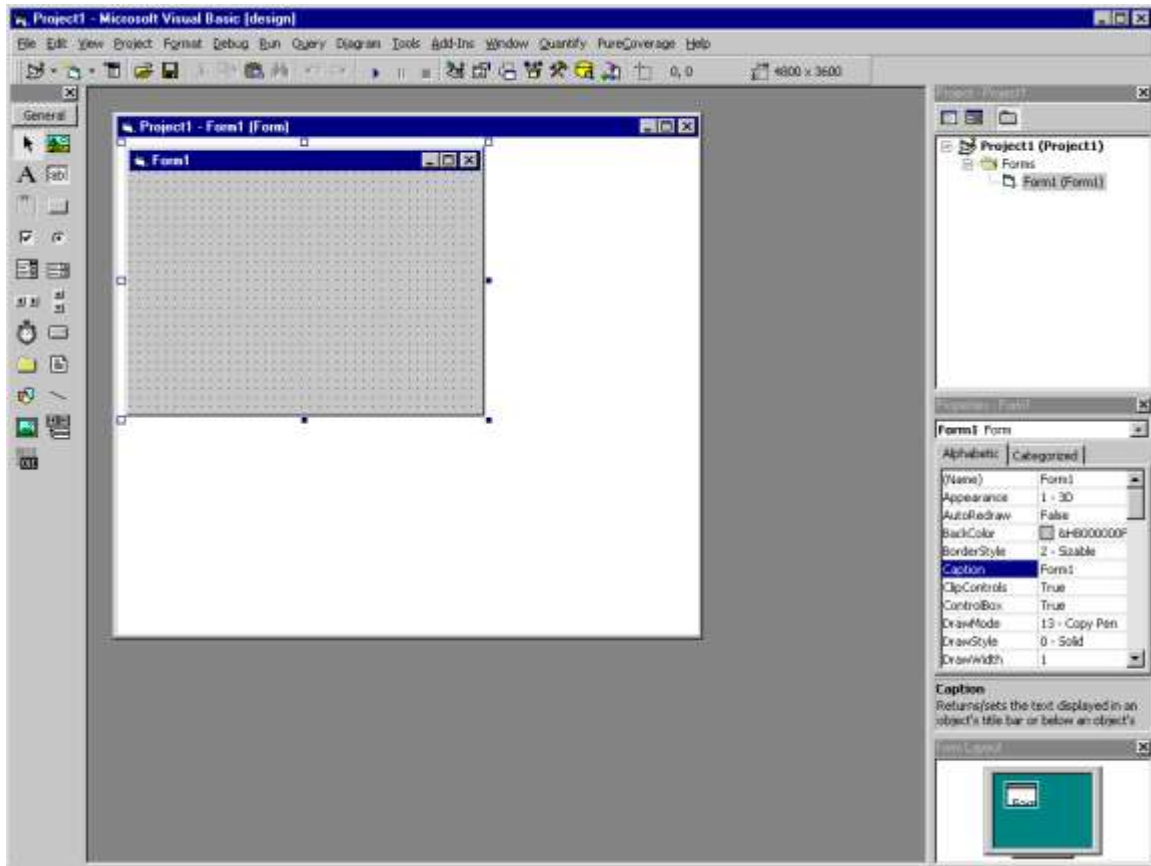
Figure 3 Visual Basic Development Environment

4. To set up the VBDisplay form you will use the Label, Textbox, and Timer controls as shown in Figure 4.
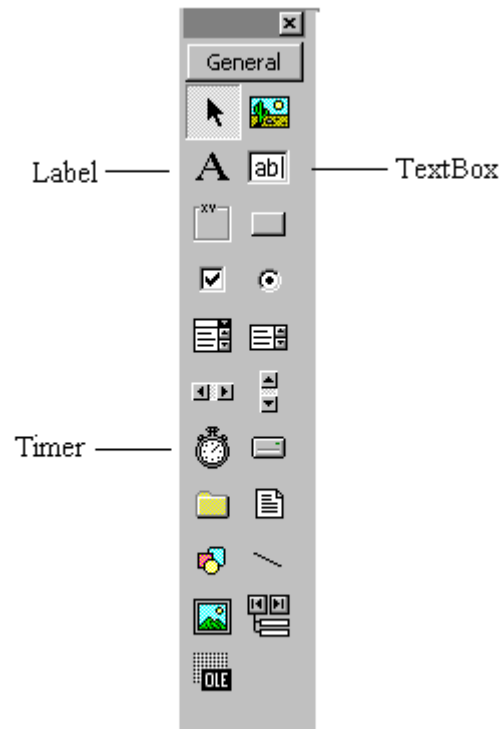
Figure 4 Toolbox Palette

5. Select the Textbox control and place one Textbox on the form window. Then copy and paste the Textbox twice until you have three Textboxes on your form. Your form should look like the one shown in Figure 5. While you are pasting the Textboxes, you might receive a warning asking you if you want to create an array. If you receive this message, choose no and continue pasting.
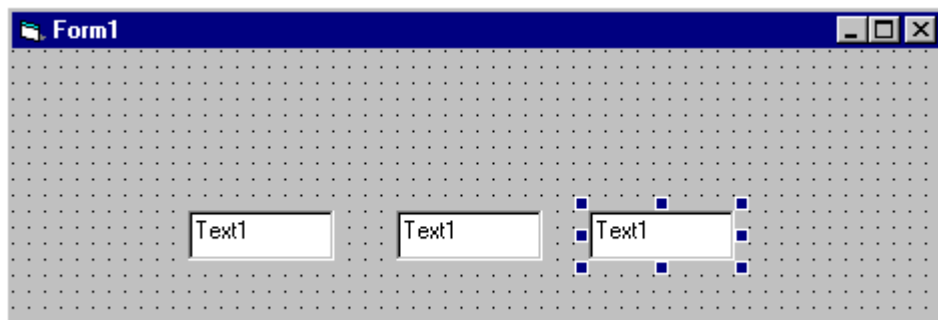
Figure 5 Form Window with Textboxes added

6. Select the first Textbox. Go to the Properties window and find the property entitled (Name). The (Name) property is at the top of the alphabetic list. Change the first Textbox's Name property to MSID038Value, the second Textbox's Name property to MSID038Status, and the third Textbox's Name property to MSID038APIReturn.
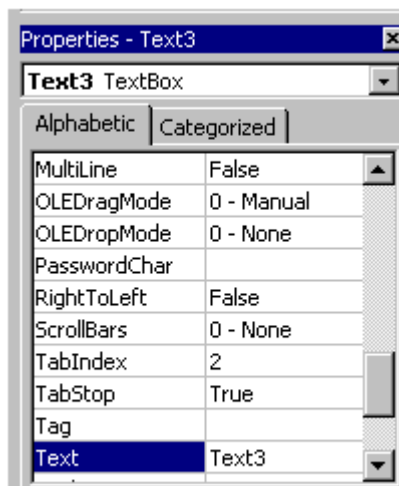
Figure 6 Properties Window for the Text boxes

7. This is a good place to stop and save your work. Use the Windows NT Explorer to create a New folder for your Visual Basic program. Name the folder VBDisplay. Go to the **File** menu and select **Save Project**. Visual Basic will prompt you to save your form. The default name for the form will be Form1.frm. Change the name to VBDisplay.frm. Locate the VBDisplay folder and push **Save**.
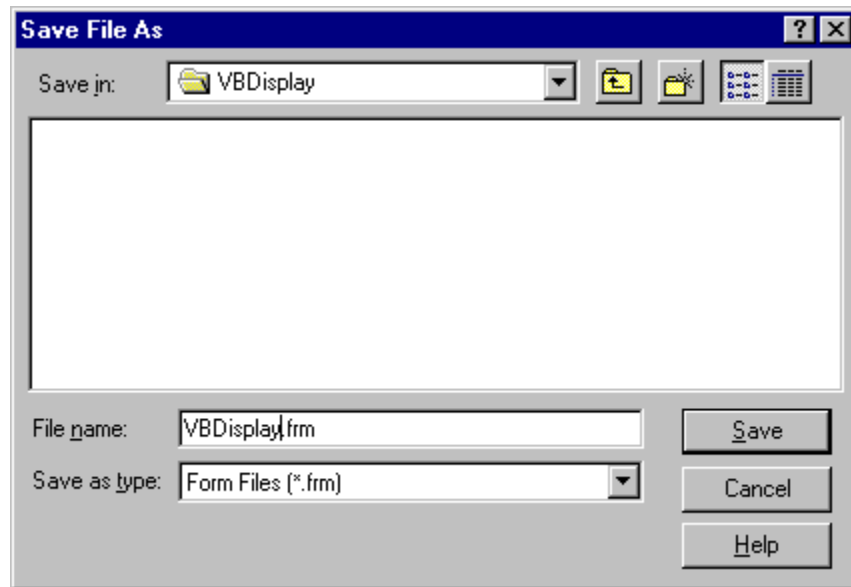
Figure 7 Save File As Dialog

8. Visual Basic will now prompt you to save the Project as shown in Figure 8. Enter the name VBDisplay.vbp and select **Save**.
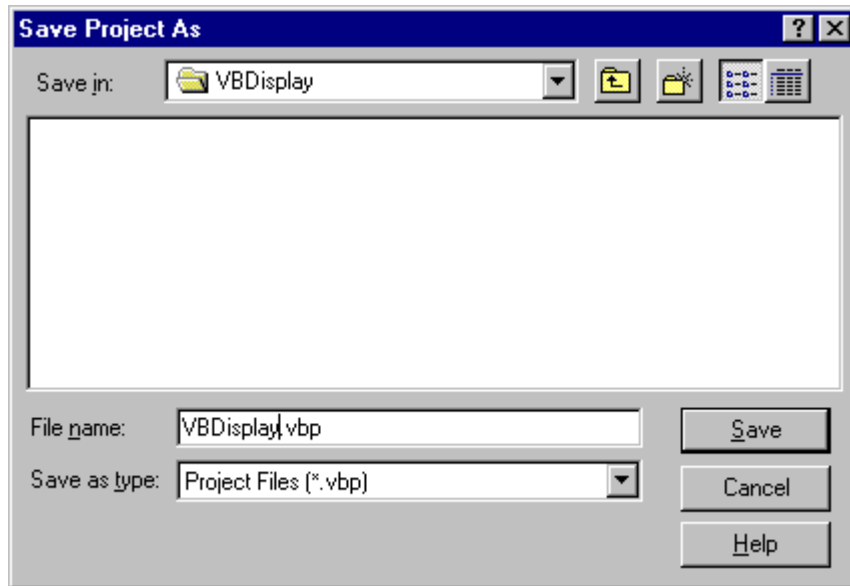
Figure 8 Save Project As Dialog

9. Each Textbox control on the form needs a corresponding Label. Using the Label tool from the Toolbox palette, place 4 labels on the form as shown in Figure 9. When you first place the labels on the form, they will be called Label1, Label2, etc. Select the label control that is located next to the Textbox control on the far left side of the form. In the Properties window find the property called Caption. Change the caption to **MSID038:**. Follow the same procedure for the other three labels. The captions should be **Value**, **Status**, and **API Return** as shown in Figure 9.
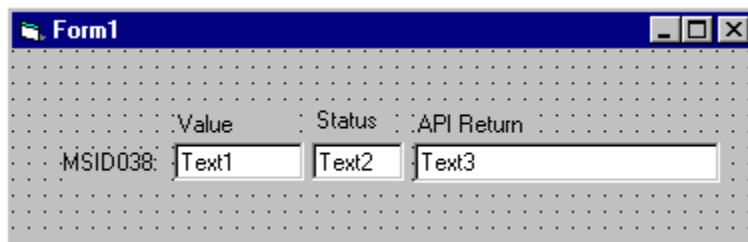
Figure 9 Labels Added to Form

8

10. The last control to add to the form is the Timer control. The Timer control makes the display cyclic. You will only see the Timer control when you are in development mode. When you run the program you won't see the Timer control. Choose the Timer control from the Toolbox palette and place it in the upper left corner of the form as in Figure 10.
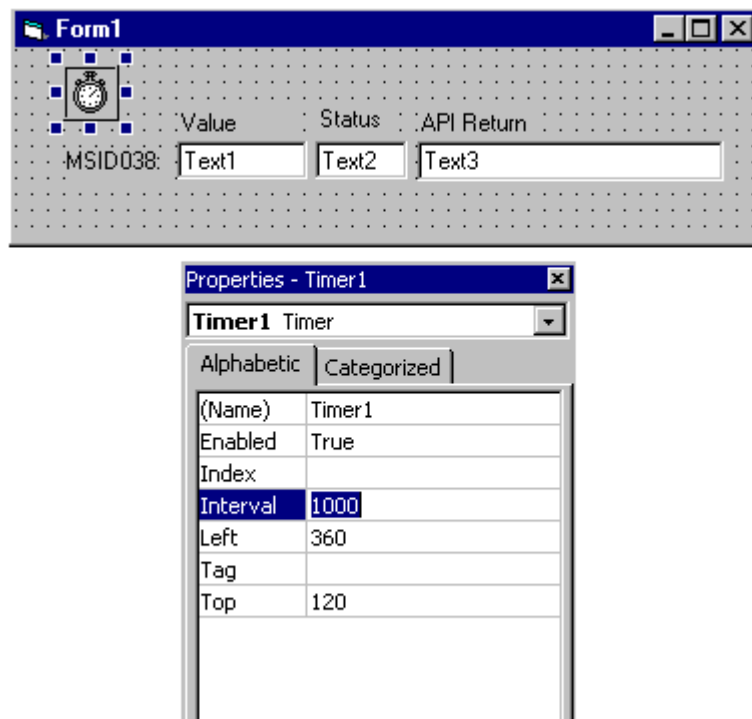
Figure 10 Timer Added to Form

11. Click on the Timer control and move to the Properties window. Find the item named Interval and click in the property field to the right of it. Set the Interval to 1000. Your display is now set to update every 1000 milliseconds.

12. You have now completed building the form portion of the Visual Basic program. Save your work by selecting **Save Project** from the **File** menu.

13. It's now time to add the code that will make the program do something interesting. But before you do that you need to make sure Visual Basic knows how to access the TReK Application Programming Interface (API) Library. To do this you need to include the trek_user_api.bas file in your VBDisplay project. The trek_user_api.bas file contains multiple Declare statements that tell Visual Basic how to find the TReK API Dynamic Linked Library. This file also contains the function prototypes in the form of Declare statements for each of the TReK API functions that are available for use with Visual Basic. You need to copy the trek_user_api.bas file into your VBDisplay directory. Go to the TReK installation directory. You will find the trek_user_api.bas file in the lib directory. Copy the trek_user_api.bas file into your VBDisplay directory.

14. Now that you have the trek_user_api.bas file in your VBDisplay directory you need to include it in your VBDisplay project. Go to the Visual Basic **Project** menu and select **Add File…**. The Add File dialog box is shown in Figure 11. Select trek_user_api.bas and push **Open**.
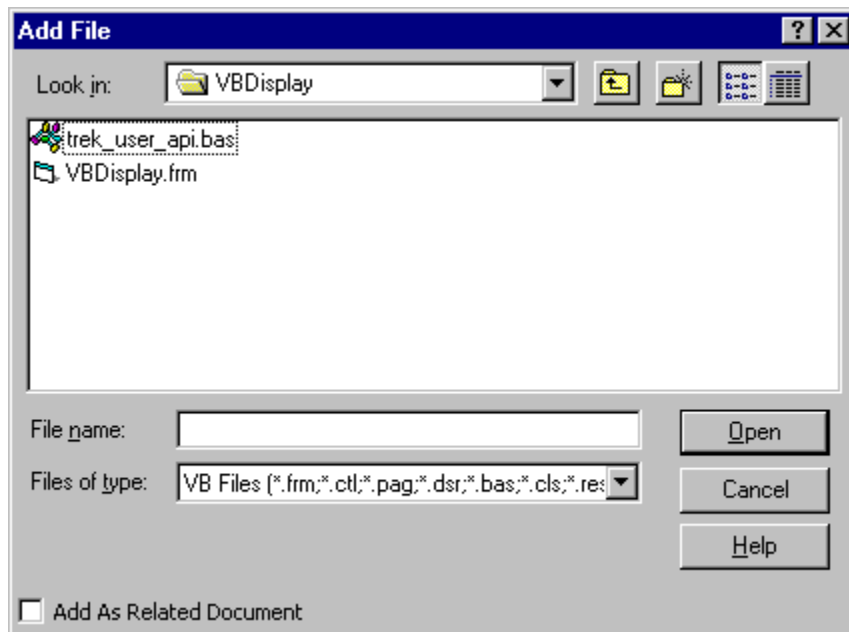
Figure 11 Add File Dialog Box

10

15. The trek_user_api.bas file is added to the VBDisplay project as shown in Figure 12.
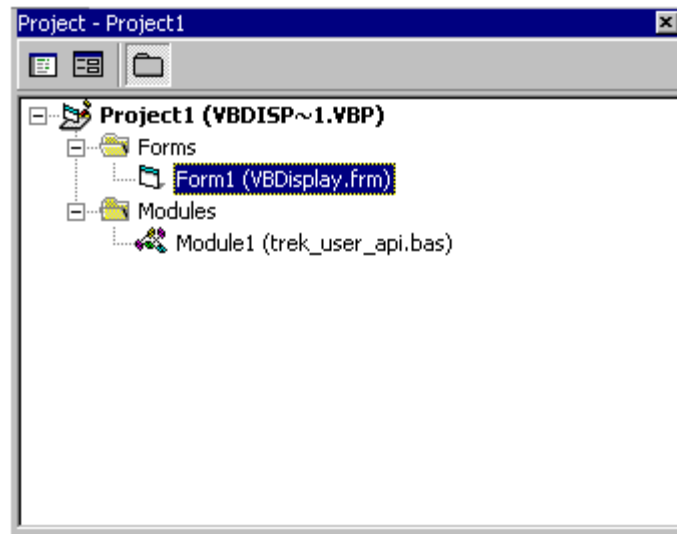
Figure 12 Module Included in Project

16. Save the program by selecting **Save Project** from the **File** menu.

17. There are three more files that are needed for the display to run properly. They are MFC71.dll, msvcp71.dll, and msvcr71.dll. You need to copy these files into the folder with the display executable or into the Windows SYSTEM32 folder. The easiest is probably into the SYSTEM32 folder, because you will not have to copy the files for every project that you create. You can find these files in the TReK installation directory in the Examples\Visual Basic\Executables folder.

18. Now that Visual Basic knows how to access the TReK API, you can add the code that will make the display functional.

19. Go to the **View** menu and choose **Code**. A blank code window will be the active window. Make sure you see General and Declarations as shown in Figure 13. If not, use the drop down menus to select General and Declarations. The Declarations section will be used to set up the token that is used in the API call.
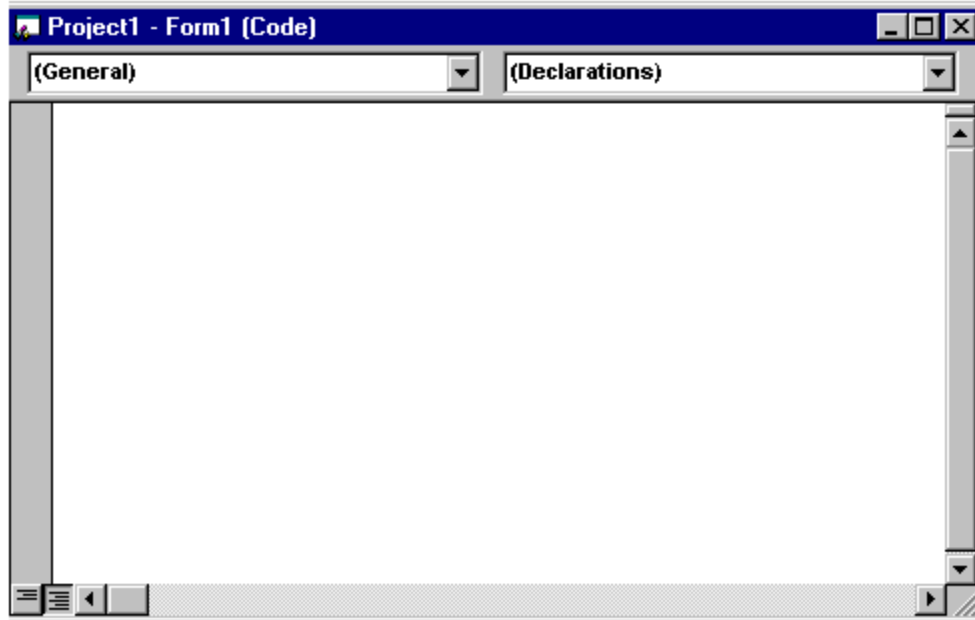
Figure 13 Blank Code Window

20. Add the following code for the Declarations section. You might want to copy and paste the code directly from the VBDisplay example program in order to save time.

```
Dim msid038_token(3) As Long
```

21. Next you need to add some code that will initialize the token when the form is loaded. Go to the form and double click on it. Visual Basic will add the Private Sub Form_Load() event procedure and take you to it. Inside this event procedure, add `token_flag = true` as shown below. (Note: The token_flag variable has been declared for you in the trek_user_api.bas file).

```
Private Sub Form_Load()

    token_flag = False

End Sub
```

22. Next you will enter the code for the Timer event procedure. This code will be executed every 1000 milliseconds. Go to the form window and double click on the Timer control. You will be taken to the Code window where you can input the following code for the `Private Sub Timer1_Timer()` event procedure.

```
Private Sub Timer1_Timer()

  Dim code As Long
  Dim return_value As Long
  Dim msid038_value As Long
  Dim msid038_status As String
  Dim api_return_string As String

  ' If token_flag is False, which it will be the first
  ' time through the loop, then initialize the token.
  ' This should only be done once.  Set the token_flag
  ' to True at the end of the loop.
  If (token_flag = False) Then
      msid038_token(0) = 0
      msid038_token(1) = 0
      msid038_token(2) = 0
      token_flag = True
  End If

  'Allocate memory for the status string buffer.
  msid038_status = String$(8, 0)
  api_return_string = String$(70, 0)

  return_value = GetOneNewestConvertedIntegerValue(PDSS_PAYLOAD,
  "MSID038", "", REAL_TIME, NO_SENSE, msid038_token(0),
  msid038_value, msid038_status)
   code = GetAPIReturnCodeAsString(return_value, 70,
  api_return_string)
  MSID038Value.Text = msid038_value
  MSID038Status.Text = msid038_status
  MSID038APIReturn.Text = api_return_string

End Sub
```

13

23. It's a good time to save your work. Select **Save Project** from the **File** menu.

24. Since you have completed the VBDisplay program you can now run the program. If you try to run the program before you start the Telemetry Processing application you will receive the error message shown in Figure 14.
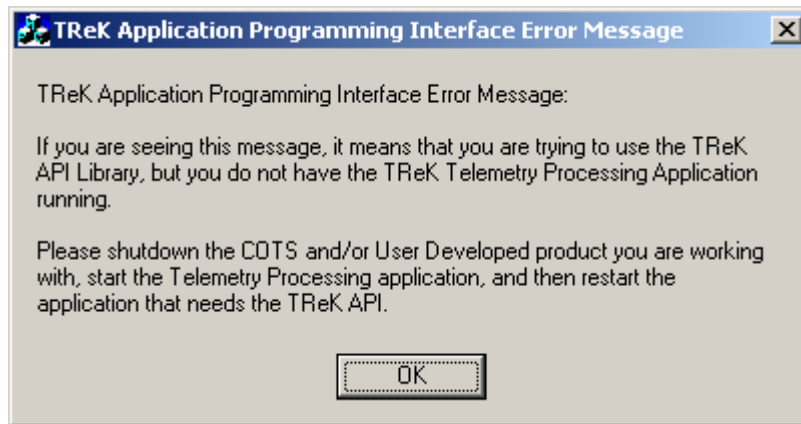
Figure 14 TReK Application Programming Interface Error Message

25. To avoid the error message above perform the following steps:

1. Start the Telemetry Processing application.
2. Add Packet ID 7 to the packet list and activate it (Packet Type = PDSS Payload, Data Mode = Real Time).
3. Start the Training Simulator application.
4. Add Packet ID 7 to the list and set the Run Time to 180 seconds.
5. Once the Telemetry Processing application finishes activating Packet ID 7, use the Training Simulator application to Send Packet ID 7.

Once that is complete, you can run your display. As long as the Training Simulator application is sending data, you should see MSID038 update with a new value once a second.

Appendix A Glossary

Note: This Glossary is global to all TReK documentation. All entries listed may not be referenced within this document.

Application Programming Interface (API)	A set of functions used by an application program to provide access to a system's capabilities.
Application Process Identifier (APID)	An 11-bit field in the CCSDS primary packet header that identifies the source-destination pair for ISS packets. The type bit in the primary header tells you whether the APID is a payload or system source-destination.
Calibration	The transformation of a parameter to a desired physical unit or text state code.
Communications Outage Recorder	System that captures and stores payload science, health and status, and ancillary data during TDRSS zone of exclusion.
Consultative Committee for Space Data Systems (CCSDS) format	Data formatted in accordance with recommendations or standards of the CCSDS.
Consultative Committee for Space Data Systems (CCSDS) packet	A source packet comprised of a 6-octet CCSDS defined primary header followed by an optional secondary header and source data, which together may not exceed 65535 octets.
Conversion	Transformation of downlinked spacecraft data types to ground system platform data types.
Custom Data Packet	A packet containing a subset of parameters that can be selected by the user at the time of request.
Cyclic Display Update Mode	A continuous update of parameters for a particular display.
Decommutation (Decom)	Extraction of a parameter from telemetry.
Discrete Values	Telemetry values that have states (e.g., on or off).

Dump

During periods when communications with the spacecraft are unavailable, data is recorded onboard and played back during the next period when communications resume. This data, as it is being recorded onboard, is encoded with an onboard embedded time and is referred to as dump data.

Enhanced HOSC System (EHS)

Upgraded support capabilities of the HOSC systems to provide multi-functional support for multiple projects. It incorporates all systems required to perform data acquisition and distribution, telemetry processing, command services, database services, mission support services, and system monitor and control services.

Exception Monitoring

A background process capable of continuously monitoring selected parameters for Limit or Expected State violations. Violation notification is provided through a text message.

Expected State Sensing

Process of detecting a text state code generator in an off-nominal state.

EXPRESS

An EXPRESS Rack is a standardized payload rack system that transports, stores and supports experiments aboard the International Space Station. EXPRESS stands for EXpedite the PRocessing of Experiments to the Space Station.

File transfer protocol (ftp)

Protocol to deliver file-structured information from one host to another.

Flight ancillary data

A set of selected core system data and payload health and status data collected by the USOS Payload MDM, used by experimenters to interpret payload experiment results.

Grayed out   Refers to a menu item that has been made insensitive, which is visually shown by making the menu text gray rather than black. Items that are grayed out are not currently available.

Greenwich Mean Time (GMT)   The solar time for the meridian passing through Greenwich, England. It is used as a basis for calculating time throughout most of the world.

Ground ancillary data   A set of selected core system data and payload health and status data collected by the POIC, which is used by experimenters to interpret payload experiment results. Ground Ancillary Data can also contain computed parameters (pseudos).

Ground receipt time   Time of packet origination. The time from the IRIG-B time signal received.

Ground Support Equipment (GSE)   GSE refers to equipment that is brought in by the user (i.e. equipment that is not provided by the POIC).

Ground Support Equipment Packet   A CCSDS Packet that contains data extracted from any of the data processed by the Supporting Facility and the format of the packet is defined in the Supporting Facility's telemetry database.

Huntsville Operations Support Center (HOSC)   A facility located at the Marshall Space Flight Center (MSFC) that provides scientists and engineers the tools necessary for monitoring, commanding, and controlling various elements of space vehicle, payload, and science experiments. Support consists of real-time operations planning and analysis, inter- and intra-center ground operations coordination, facility and data system resource planning and scheduling, data systems monitor and control operations, and data flow coordination.

IMAQ ASCII	A packet type that was added to TReK to support a very specific application related to NASA's Return to Flight activities. It is not applicable to ISS. It is used to interface with an infrared camera that communicates via ASCII data.
Limit Sensing	Process of detecting caution and warning conditions for a parameter with a numerical value.
Line Outage Recorder Playback	A capability provided by White Sands Complex (WSC) to play back tapes generated at WSC during ground system communication outages.
Measurement Stimulus Identifier (MSID)	Equivalent to a parameter.
Monitoring	A parameter value is checked for sensing violations. A message is generated if the value is out of limits or out of an expected state.
Parameter	TReK uses the generic term parameter to mean any piece of data within a packet. Sometimes called a measurement or MSID in POIC terminology.
Payload Data Library (PDL)	An application that provides the interface for the user to specify which capabilities and requirements are needed to command and control his payload.
Payload Data Services Systems (PDSS)	The data distribution system for ISS. Able to route data based upon user to any of a number of destinations.
Payload Health and Status Data	Information originating at a payload that reveals the payload's operational condition, resource usage, and its safety/anomaly conditions that could result in damage to the payload, its environment or the crew.
Payload Operations Integration Center (POIC)	Manages the execution of on-orbit ISS payloads and payload support systems in coordination/unison with distributed International Partner Payload Control Centers, Telescience Support Centers (TSC's) and payload-unique remote facilities.

Payload Rack Checkout Unit (PRCU)
: The Payload Rack Checkout Unit is used to verify payload to International Space Station interfaces for U.S. Payloads.

Playback
: Data retrieved from some recording medium and transmitted to one or more users.

Pseudo Telemetry (pseudo data)
: Values that are created from calculations instead of directly transported telemetry data. This pseudo data can be created from computations or scripts and can be displayed on the local PC.

Remotely Generated Command
: A command sent by a remote user whose content is in a raw bit pattern format. The commands differ from predefined or modifiable commands in that the content is not stored in the POIC Project Command Database (PCDB).

Science data
: Sensor or computational data generated by payloads for the purpose of conducting scientific experiments.

Subset
: A collection of parameters from the total parameter set that is bounded as an integer number of octets but does not constitute the packet itself. A mini-packet.

Super sampled
: A parameter is super sampled if it occurs more than once in a packet.

Swap Type
: A flag in the Parameter Table of the TReK database that indicates if the specified datatype is byte swapped (B), word swapped (W), byte and word swapped (X), byte reversal (R), word reversal (V) or has no swapping (N).

Switching
: A parameter's value can be used to switch between different calibration and sensing sets. There are two types of switching on TReK: range and state code.

Transmission Control Protocol (TCP)	TCP is a connection-oriented protocol that guarantees delivery of data.
Transmission Control Protocol (TCP) Client	A TCP Client initiates the TCP connection to connect to the other party.
Transmission Control Protocol (TCP) Server	A TCP Server waits for (and accepts connections from) the other party.
Telemetry	Transmission of data collected form a source in space to a ground support facility. Telemetry is downlink only.
Telescience Support Center (TSC)	A TSC is a NASA funded facility that provides the capability to plan and operate on-orbit facility class payloads and experiments, other payloads and experiments, and instruments.
User Application	Any end-user developed software program that uses the TReK Application Programming Interface software. Used synonymously with User Product.
User Data Summary Message (UDSM)	Packet type sent by PDSS that contains information on the number of packets sent during a given time frame for a PDSS Payload packet. For details on UDSM packets, see the POIC to Generic User IDD (SSP-50305).
Uplink format	The bit pattern of the command or file uplinked.
User Datagram Protocol (UDP)	UDP is a connection-less oriented protocol that does not guarantee delivery of data. In the TCP/IP protocol suite, the UDP provides the primary mechanism that application programs use to send datagrams to other application programs. In addition to the data sent, each UDP message contains both a destination port number and a fully qualified source and destination addresses making it possible for the UDP software on the destination to deliver the message to the correct recipient process and for the recipient process to send a reply.

User Product   Any end-user developed software program that
                                      uses the TReK Application Programming Interface
                                      software. Used synonymously with User
                                      Application.

Web   Term used to indicate access via HTTP protocol;
                                      also referred to as the World Wide Web (WWW).

Appendix B Acronyms

Note: This acronym list is global to all TReK documentation. Some acronyms listed may not be referenced within this document.

AOS	Acquisition of Signal
API	Application Programming Interface
APID	Application Process Identifier
ASCII	American Standard Code for Information Interchange
CAR	Command Acceptance Response
CAR1	First Command Acceptance Response
CAR2	Second Command Acceptance Response
CCSDS	Consultative Committee for Space Data Systems
CDB	Command Database
CDP	Custom Data Packet
COR	Communication Outage Recorder
COTS	Commercial-off-the-shelf
CRR	Command Reaction Response
DSM	Data Storage Manager
EHS	Enhanced Huntsville Operations Support Center (HOSC)
ERIS	EHS Remote Interface System
ERR	EHS Receipt Response
EXPRESS	Expediting the Process of Experiments to the Space Station
ES	Expected State
FAQ	Frequently Asked Question
FDP	Functionally Distributed Processor
FSV	Flight System Verifier
FSV1	First Flight System Verifier
FSV2	Second Flight System Verifier
FPD	Flight Projects Directorate
FTP	File Transfer Protocol
GMT	Greenwich Mean Time
GRT	Ground Receipt Time
GSE	Ground Support Equipment
HOSC	Huntsville Operations Support Center
ICD	Interface Control Document
IMAQ ASCII	Image Acquisition ASCII
IP	Internet Protocol
ISS	International Space Station
LDP	Logical Data Path
LES	Limit/Expected State
LOR	Line Outage Recorder
LOS	Loss of Signal
MCC-H	Mission Control Center – Houston
MOP	Mission, Operational Support Mode, and Project
MSFC	Marshall Space Flight Center

MSID	Measurement Stimulus Identifier
NASA	National Aeronautics and Space Administration
OCDB	Operational Command Database
OS	Operating System
PC	Personal Computer, also Polynomial Coefficient
PCDB	POIC Project Command Database
PDL	Payload Data Library
PDSS	Payload Data Services System
PGUIDD	POIC to Generic User Interface Definition Document
POIC	Payload Operations Integration Center
PP	Point Pair
PRCU	Payload Rack Checkout Unit
PSIV	Payload Software Integration and Verification
RPSM	Retrieval Processing Summary Message
SC	State Code
SCS	Suitcase Simulator
SSP	Space Station Program
SSCC	Space Station Control Center
SSPF	Space Station Processing Facility
TCP	Transmission Control Protocol
TReK	Telescience Resource Kit
TRR	TReK Receipt Response
TSC	Telescience Support Center
UDP	User Datagram Protocol
UDSM	User Data Summary Message
URL	Uniform Resource Locator
USOS	United States On-Orbit Segment
VCDU	Virtual Channel Data Unit
VCR	Video Cassette Recorder
VPN	Virtual Private Network